EPJ Web of Conferences **175**, 06014 (2018) https://doi.org/10.1051/epjconf/201817506014
Lattice 2017

Baryonic and mesonic 3-point functions with open spin indices

Gunnar S. Bali[1,2], *Sara* Collins[1], *Benjamin* Gläßle[1], *Simon* Heybrock[3], *Piotr* Korcyl[1,4], *Marius* Löffler[1,⋆], *Rudolf* Rödl[1], and *Andreas* Schäfer[1]

[1] *Institut für Theoretische Physik, Universität Regensburg, D-93040 Regensburg, Germany*
[2] *Department of Theoretical Physics, Tata Institute of Fundamental Research, Homi Bhabha Road, Mumbai 400005, India*
[3] *European Spallation Source, 225 92 Lund, Sweden*
[4] *M. Smoluchowski Institute of Physics, Jagiellonian University, ul. Łojasiewicza 11, 30-348 Kraków, Poland*

Abstract. We have implemented a new way of computing three-point correlation functions. It is based on a factorization of the entire correlation function into two parts which are evaluated with open spin- (and to some extent flavor-) indices. This allows us to estimate the two contributions simultaneously for many different initial and final states and momenta, with little computational overhead. We explain this factorization as well as its efficient implementation in a new library which has been written to provide the necessary functionality on modern parallel architectures and on CPUs, including Intel's Xeon Phi series.

1 Introduction

Observables constructed with the use of three-point correlation functions can describe a multitude of physical phenomena, such as the parton structure of hadrons or their weak transitions, depending on the operator type at the insertion. The relevant strong interaction matrix elements can be computed using Lattice Quantum Chromodynamics. Traditionally, a method employed to that aim was the sequential source method [1]. The numerical cost of this is high because a new inversion is necessary for each final momentum at the sink. In this contribution we present a stochastic algorithm which circumvents this limitation and disentangles the number of inversions of the lattice Dirac operator from the number of sink/insertion momenta. The implementation we propose parallelizes the computations in such a way that multiple source positions and multiple insertion positions can be estimated simultaneously. Moreover, by storing the uncontracted data, with all spin indices open, on disk, we enable the user to analyse any channel of interest at a later stage.

An aspect of our implementation which we wish to highlight in this contribution is the extensive use of vectorization. As the compute power of today's processors relies on longer and longer vector registers of additional vector processing units, an adequate data layout is required to efficiently use these resources. We explain our data layout which supports large vector registers, therefore enabling us to efficiently run our code on the Intel's Xeon Phi processors featuring 512 bit AVX vector instructions.

⋆Speaker, e-mail: marius.loeffler@physik.uni-regensburg.de

Our implementation is based on a framework developed specifically for Intel's Xeon Phi processors, called `LibHadronAnalysis` see, e.g., [2, 3]. This library contains additional routines for computing meson and baryon spectra, meson and baryon distribution amplitudes and many other objects. Compared to a naive implementation in Chroma [4] using QDP++ [4] objects it provides speed-up factors of the order 10-20 on the KNC and KNL architectures.

Below we describe in detail the algorithm. In particular we explain how the computation of a three-point correlation function can be separated into two, largely independent parts, the "spectator" and the "insertion" parts. We pay special attention to the parallelization schemes which are different for each of these parts. Subsequently, we present benchmarks showing the performance of our implementation on a KNL cluster, and then we conclude.

2 Stochastic baryonic and mesonic three-point correlation functions

In this section we introduce the three-point correlation function we are interested in and show how it can be factorized into two, largely independent parts. We only show explicit formulae for the case of meson three-point functions, for the sake of notational simplicity. A similar approach was used in Refs. [5–8], however, here we keep all spin indices open.

2.1 General structure

A three-point meson correlation function (c.f., figure 1) with a source $C(r)$, a sink $A(x')$ and an insertion operator $I(y)$, located at timeslices r_4, x'_4 and y_4 respectively, reads

$$\begin{aligned}\langle A(x')\, I(y)\, C(r)\rangle &= \mathrm{tr}\left[G_{f_1}(r,x')\;\Gamma_{\mathrm{snk}}\;G_{f_2}(x',y)\;\Gamma_{\mathrm{ins}}\;G_{f_3}(y,r)\;\Gamma_{\mathrm{src}}\right] \\ &= \delta_{ab}\;\delta_{a'b'}\;\delta_{\tilde{a}\tilde{b}}\;\Gamma^{\alpha'\beta'}_{\mathrm{snk}}\;\Gamma^{\tilde{\alpha}\tilde{\beta}}_{\mathrm{ins}}\;\Gamma^{\beta\alpha}_{\mathrm{src}}\;G_{f_1}(r,x')^{\alpha\alpha'}_{aa'}\;G_{f_2}(x',y)^{\beta'\tilde{\alpha}}_{b'\tilde{a}}\;G_{f_3}(y,r)^{\tilde{\beta}\beta}_{\tilde{b}b} \qquad (1)\end{aligned}$$

where

$$A(\mathbf{x}',x'_4) = \delta_{a'b'}\;\overline{\psi}_{f_2}(\mathbf{x}',x'_4)^{\alpha'}_{a'}\;\Gamma^{\alpha'\beta'}_{\mathrm{snk}}\;\psi_{f_1}(\mathbf{x}',x'_4)^{\beta'}_{b'}\;, \qquad (2)$$

$$C(\mathbf{r},r_4) = \delta_{ba}\;\overline{\psi}_{f_1}(\mathbf{r},r_4)^{\beta}_{b}\left(\gamma_4\Gamma^{\dagger}\gamma_4\right)^{\beta\alpha}\;\psi_{f_3}(\mathbf{r},r_4)^{\alpha}_{a}\;, \qquad (3)$$

$$I(\mathbf{y},y_4) = \delta_{\tilde{a}\tilde{b}}\;\overline{\psi}_{f_3}(\mathbf{y},y_4)^{\tilde{\alpha}}_{\tilde{a}}\;\Gamma^{\tilde{\alpha}\tilde{\beta}}_{\mathrm{ins}}\;\psi_{f_2}(\mathbf{y},y_4)^{\tilde{\beta}}_{\tilde{b}}\;, \qquad (4)$$

are the annihilation, creation and insertion operators respectively, with $f_i \in \{l,s,c\}$ (light, strange, charm). $G_{f_i}(r,x)$ is a standard fermion propagator from x to r of flavor f_i. We use the convention to denote the annihilation spin and color operator indices with primed Greek and Latin letters α', a', creation operator indices with ordinary Greek and Latin letters α, a, and the insertion operator indices with tilde Greek and Latin letters $\tilde{\alpha},\tilde{a}$. Γ_{ins} can contain local derivatives and A and C may contain quark smearing.

At this point we replace one of the propagators by its stochastic estimate

$$G_{f_2}(y,x')^{\tilde{\alpha}\beta'}_{\tilde{a}b'} \approx \frac{1}{N}\sum_{i=1}^{N} s_{i,f_2}(y)^{\tilde{\alpha}}_{\tilde{a}}\;(\eta_i^*)\,(x')^{\beta'}_{b'}\;, \qquad (5)$$

where the sum runs over N realizations of the noise source vector $\eta_i(x')$, with the properties

$$\frac{1}{N}\sum_{i=1}^{N} (\eta_i)(x)_a^{\alpha} = 0 + O\left(\frac{1}{\sqrt{N}}\right) , \tag{6}$$

$$\frac{1}{N}\sum_{i=1}^{N} (\eta_i)(x)_a^{\alpha} (\eta_i^*)(x')_{a'}^{\alpha'} = \delta_{xx'}\ \delta_{\alpha\alpha'}\ \delta_{aa'} + O\left(\frac{1}{\sqrt{N}}\right) . \tag{7}$$

The $(\eta_i)(x)$ are time partitioned and set to zero, unless $x_4 = x_4'$ or $x_4 = x_4''$.

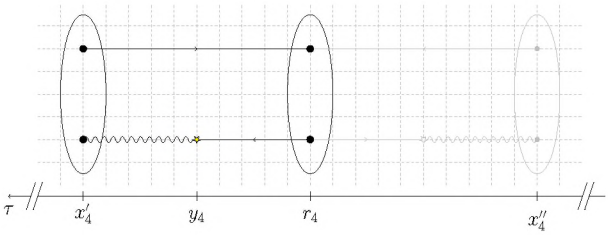

Figure 1. Sketch of the structure of a generic three-point correlation function.

In figure 1 we show the generic structure of a three-point correlation function in the meson case. The central ellipse denotes a meson created with operator Eq. (3) in the middle of the lattice at timeslice r_4. The meson is annihilated by the operator from Eq. (2) at the timeslice x_4' as depicted by the left-most ellipse. Arrows denote exact point-to-all propagators. The star at timeslice y_4 denotes one of the possible positions of the insertion operator. The wiggly line is used to plot the stochastic all-to-all propagator. We call these four elements the "forward" correlation function as opposed to the shaded mirror-reflected graph on the right hand side which corresponds to the "backward" process. The forward and backward diagrams are estimated simultaneously which allows for increased statistics.

The introduction of the stochastic propagator allows us to factorize the correlation function $\langle A(x')\, I(y)\, C(r)\rangle$ into two parts [5] as follows

$$\begin{aligned}\langle A(x')\, I(y)\, C(r)\rangle = \ & \delta_{ab}\ \delta_{a'b'}\ \delta_{\check{a}\check{b}}\ \Gamma_{\text{snk}}^{\alpha'\beta'}\ \Gamma_{\text{ins}}^{\check{\alpha}\check{\beta}}\ \Gamma_{\text{src}}^{\beta\alpha} \times \\ & \times \underbrace{\left[\gamma_5\ G_{f_1}^{\dagger}(x', r)\ \gamma_5\right]_{a'a}^{\alpha'\alpha}\ \left[\eta_i(x')\ \gamma_5\right]_{b'}^{\beta'}}_{\triangleq \text{Spectator}}\ \underbrace{\left[\gamma_5\ s_{i,f_2}(y)\right]_{\check{a}}^{*\,\check{\alpha}}\ G_{f_3}(y, r)_{\check{b}b}^{\check{\beta}\beta}}_{\triangleq \text{Insertion}}\end{aligned} \tag{8}$$

We define the spectator $S_{i,f_1}(\mathbf{p}, x_4')_a^{\beta'\,\alpha'\,\alpha}$ and the insertion $I_{i,f_2,f_3}(\mathbf{q}, y_4)_a^{\check{\alpha}\check{\beta}\beta}$ parts

$$S_{i,f_1}(\mathbf{p}', x_4')_a^{\beta'\,\alpha'\,\alpha} = \sum_{\mathbf{x}'} \delta_{a'b'}\ \left[\eta_i(x')\ \gamma_5\right]_{b'}^{\beta'} \left[\gamma_5\ G_{f_1}^{\dagger}(x', r)\ \gamma_5\right]_{a'a}^{\alpha'\alpha} \cdot e^{-i\mathbf{p}'\cdot\mathbf{x}'} , \tag{9}$$

$$I_{i,f_2,f_3}(\mathbf{q}, y_4)_a^{\check{\alpha}\check{\beta}\beta} = \sum_{\mathbf{y}} \delta_{ab}\ \delta_{\check{a}\check{b}}\ \left[\gamma_5\ s_{i,f_2}(y)\right]_{\check{a}}^{*\,\check{\alpha}}\ G_{f_3}(y, r)_{\check{b}b}^{\check{\beta}\beta} \cdot e^{i\mathbf{q}\cdot\mathbf{y}} , \tag{10}$$

where we have assumed $\mathbf{r} = 0$. Otherwise we have to replace $\mathbf{x}' \to \mathbf{x}' - \mathbf{r}$, $\mathbf{y} \to \mathbf{y} - \mathbf{r}$.

2.2 Spectator part

The computation of the spectator part consists of the contractions of propagators at the timeslices where the source and the sink are located. Naively only the MPI ranks working on the timeslices r_4

and x_4', x_4'' would work. In our implementation we prepare a set of propagators sourced from different temporal source positions, as shown on figure 2. We redistribute these propagators among the different MPI ranks in such a way that each rank has at least one propagator. The computation of the spectator part for each source position is than performed simultaneously. The Fourier transformation Eq. (9) fixes the momentum $\mathbf{p}'$ at the sink.

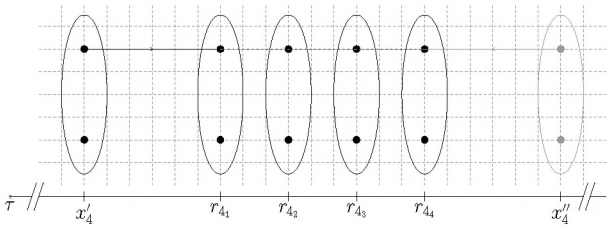

Figure 2. Parallelization of the spectator part of the three-point correlation function. Propagators sourced at different timeslices denoted by different solid ellipses are redistributed among all MPI ranks so that each rank has at least one set of propagators to work with.

2.3 Insertion part

The insertion part corresponds to the contraction of the stochastic propagator, i.e., the solution of the lattice Dirac equation sourced by random noise vectors, with the point-to-all propagator. This has to be repeated for each position y_4 of the insertion operator between the sinks x_4' and x_4 and the source r_4. Again, in order to keep a good workload balance we redistribute the data from the timeslices where the insertion is present, denoted with stars on figure 3, among all MPI ranks in such a way that each rank has approximately the same number of insertion positions to work with, and we perform all the computations in parallel. Note that a separate Fourier transformation in Eq. (10) allows to select a desired momentum $\mathbf{q}$ flowing through the insertion.

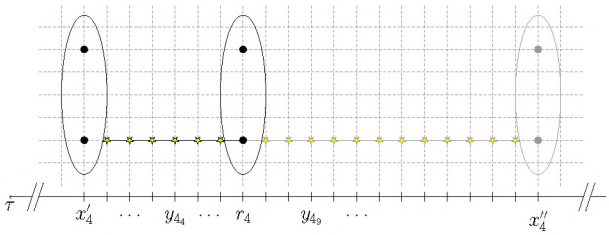

Figure 3. Parallelization of the insertion part of the three-point correlation function. Data from timeslices denoted with yellow stars is redistributed among all MPI ranks, such that all ranks have a similar workload.

3 Performance

We paid special attention to the data layout to enable the use of vectorization. The crutial point was to reorder the many loops in the algorithm. We show this explicitly for pseudocode in listings 1 and 2.

```
for 3 spin indices do
    for color indices a,b do
        for all sites in the local
          lattice do
            |
        end
    end
end
```

Algorithm 1: Reference implementation

```
for all sites in the local lattice do
    for 1 spin index do
        for color indices a,b do
            for 2 spin indices
              SIMD vect. do
                |
            end
        end
    end
end
```

Algorithm 2: LHA implementation

Benchmarks were performed on the QPACE3 supercomputer of the SFB/TR 55 at the Jülich Supercomputing Centre. The machine is based on Intel Xeon Phi (KNL) processors connected via Intel Omni-Path. We used the Intel compiler version 17.0.2.

The `LibHadronAnalysis` library [2] was incorporated into the QDP++/Chroma software stack [4], together with the multigrid solver [9–13] optimized for the KNL architecture.

The computations were carried out on a single configuration of the CLS ensemble H101 [14]. It is a $N_f = 2 + 1$ ensemble with non-perturbatively $O(a)$ improved Wilson fermions and tree-level improved Symanzik gauge action and features open boundary conditions in time. The pion and kaon masses are about 420 MeV and this $32^3 \times 96$ lattice has a lattice spacing of about 0.086 fm.

Actual measurements were performed for the meson spectator part Eq. (9) and insertion part Eq. (10) using $r_4 = 30\,a$ and a source-sink separation of $r_4 - x'_4 = x''_4 - r_4 = 10\,a$. Within the spectator part propagators are smeared at the source and the sink time-slice while in the insertion part only the source time-slice of the propagator is smeared. The number of stochastic indices is set to $N_i = 50$.

Running on 4 – 32 KNLs and distributing the 256 hardware threads on each KNL to 8 MPI tasks yields the strong scaling for the meson spectator and insertion part contraction as shown in figure 4. Note that the computation is done for 50 independently seeded stochastic estimators in forward and backward direction, i.e., the mean values are averaged over 100 computations each. The creation times for the noise and the solution vectors are not considered.

For the spectator/insertion a minimal computation time is achieved using 8/32 KNLs with 8 tasks per node. In this setup the Intel Omni-Path connection between the nodes becomes saturated, the computation is well parallelized and the overhead due to internal communication is not dominant.

In addition the wallclock-time for a particular measurement using two different ranges of momenta on one H101 configuration is evaluated – again 8 KNLs with 8 tasks per node are used. The timings for $\mathbf{k}' = \mathbf{k}^2 = 0$ and for $\mathbf{k}', \mathbf{k}^2 = 0, \ldots, 8$ for spectator and insertion momenta are shown in figure 5, where $p'_i = k'_i\, 2\pi/L$, $q_i = k_i\, 2\pi/L$ where the integers k'_i and k_i label the momentum components of $\mathbf{p}'$ and $\mathbf{q}$ within the Fourier transformation. Note that in these timings also a baryon measurement is included. In both cases the overall computation time is almost the same

`LibHadronAnalysis` wallclock-time for $\mathbf{k}'^2 = \mathbf{k}^2 = 0$: $\approx$ 530s

`LibHadronAnalysis` wallclock-time for $\mathbf{k}'^2, \mathbf{k}^2 = 0, \ldots, 8$ ($93 \cdot 93$ mom. combinations): $\approx$ 575s.

Hence it is possible to produce data for a large number of final momenta without increasing the computation time significantly. In addition the data-layout presented in Eq. (9) and Eq. (10) provides analysis capabilities for various physical channels since the Γ-structures of the source and sink interpolators are not specified during the simulations.

The $\mathbf{k}'^2, \mathbf{k}^2 = 0, \ldots, 8$ measurement was also performed using the sequential source method [1] which yields a run-time of $\approx$ 930s. Compared to the above wallclock-time of $\approx$ 575s this is a speed-up of $\approx$ 1.6 where we have not taken into account that the stochastic code gives $16 \cdot 16$ Γ-combinations at

source and sink for free. First tests have shown that the computation time of the analysis code needed to obtain final three-point function results is in the range of a few seconds and therefore is negligible.

Collectively this means that one needs $\approx$ 82 KNL core hours to perform the above measurement on a single configuration of the H101 assuming that 8 nodes with 64 cores per node are used. Altogether 2016 configurations are available to analyse the H101, i.e., at most $\approx$ 165000 KNL core hours are needed to analyze the entire ensemble for every combination of source and sink meson and baryon interpolators on a single source time slice and for the given source sink separation $\Delta t = 10\,a$. Due to the distribution shown in figure 5 the overall computation time should remain almost constant when increasing the number of source positions, always provided that the computation of the spectator part can be fully parallelized.

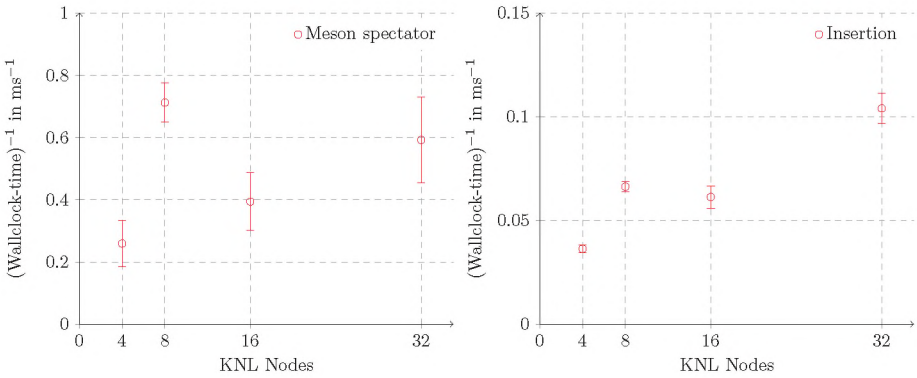

Figure 4. Strong scaling for the spectator (left) and the insertion (right) parts for the meson three-point correlation function. Measurements were done on a $32^3 \times 96$ lattice, with $\mathbf{k}^2 = \mathbf{k}'^2 = 0$ and 100 stochastic estimates for the stochastic propagators.

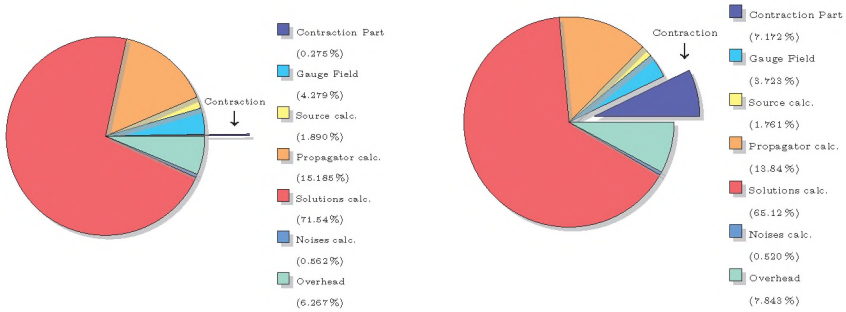

Figure 5. Contribution of the contraction time to the total time budget of the three-point correlation function estimation. On the left panel the case of total momentum 0 is shown, whereas on the right panel the spectator and insertion momentum square of $0, \ldots, 8$ (i.e., $93 \cdot 93$ momentum combinations) is shown.

4 Conclusions

The timings presented in the last section reveal that our implementation overtakes currently implemented methods at least by a factor of 1.5 in terms of computation time for high momentum square for a given source-sink structure. However, exploiting the great flexibility of `LibHadronAnalysis` output data makes it feasible to reuse the data for many source-sink Dirac Γ-matrix combinations which results in an incredible economization. Furthermore it is sufficient to compute the insertion part only once and use it in both, baryon and meson, measurements. Since it is now possible to generate data containing an enormous amount of information it is also necessary to process the data further to finally get physical observables. The analysis software package is still in development and will be released as soon as possible.

Acknowledgements

This work is funded by Deutsche Forschungsgemeinschaft (DFG) within the transregional collaborative research centre 55 (SFB-TRR55). The Chroma software suite [4] was used extensively in this work along with the multigrid solver implementation of [13]. Computations were performed on the SFB/TR55 QPACE supercomputers.

References

[1] G. Martinelli, C. Sachrajda, Nucl. Phys. B **316**, 355–372 (1989)
[2] *Lib hadron analysis*, https://rqcd.ur.de:8443/hes10653/lib-hadron-analysis (2017)
[3] S. Heybrock, M. Rottmann, P. Georg, T. Wettig, PoS **LATTICE2015**, 036 (2016)
[4] R.G. Edwards, B. Joo (SciDAC, LHPC, UKQCD), Nucl. Phys. Proc. Suppl. **140**, 832 (2005)
[5] R. Evans, G. Bali, S. Collins, Phys. Rev. **D82**, 094501 (2010)
[6] C. Alexandrou, S. Dinter, V. Drach, K. Jansen, K. Hadjiyiannakou, D.B. Renner (ETM), Eur. Phys. J. **C74**, 2692 (2014)
[7] G.S. Bali, S. Collins, B. Gläßle, M. Göckeler, J. Najjar, R. Rödl, A. Schäfer, A. Sternbeck, W. Söldner, PoS **LATTICE2013**, 271 (2014)
[8] Y.B. Yang, A. Alexandru, T. Draper, M. Gong, K.F. Liu, Phys. Rev. **D93**, 034503 (2016)
[9] D. Richtmann, S. Heybrock, T. Wettig, PoS **LATTICE2015**, 035 (2016)
[10] P. Arts, et al., CoRR **abs/1502.04025** (2015)
[11] A. Frommer, K. Kahl, S. Krieg, B. Leder, M. Rottmann, SIAM J.Sci.Comput. **36**, A1581 (2014)
[12] P. Georg, D. Richtmann, T. Wettig, PoS **LATTICE2016**, 361 (2017)
[13] P. Georg, D. Richtmann, T. Wettig, *DD-αAMG on QPACE 3* (2017)
[14] M. Bruno, et al., JHEP **2015**, 43 (2015)